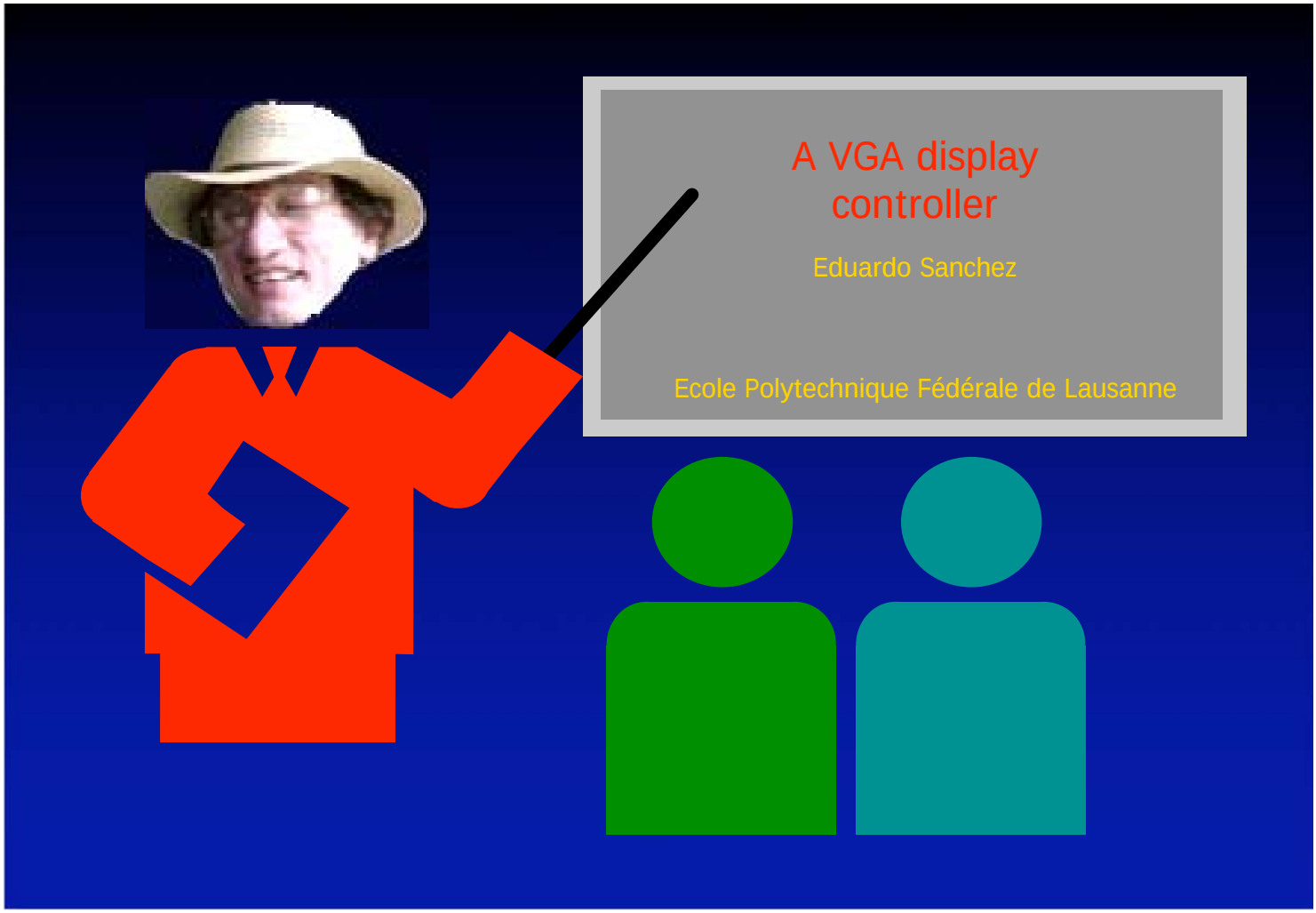
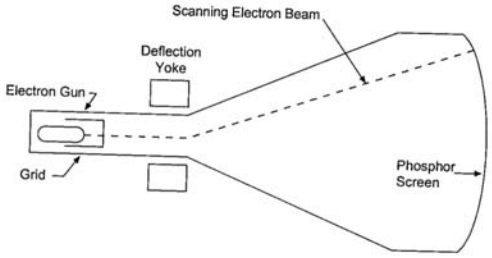
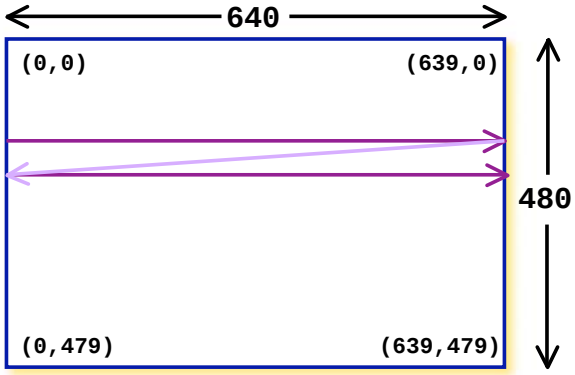


VGA video signal generation

- A VGA video signal contains 5 active signals:
  - horizontal sync: digital signal, used for synchronisation of the video
  - vertical sync: digital signal, used for synchronisation of the video
  - red (R): analog signal (0-0.7 v), used to control the color
  - green (G): analog signal (0-0.7 v), used to control the color
  - blue (B): analog signal (0-0.7 v), used to control the color
- By changing the analog levels of the three RGB signals all other colors are produced
- The electron beam must be scanned over the viewing screen in a sequence of horizontal lines to generate an image. The RGB color information in the video signal is used to control the strength of the electron beam



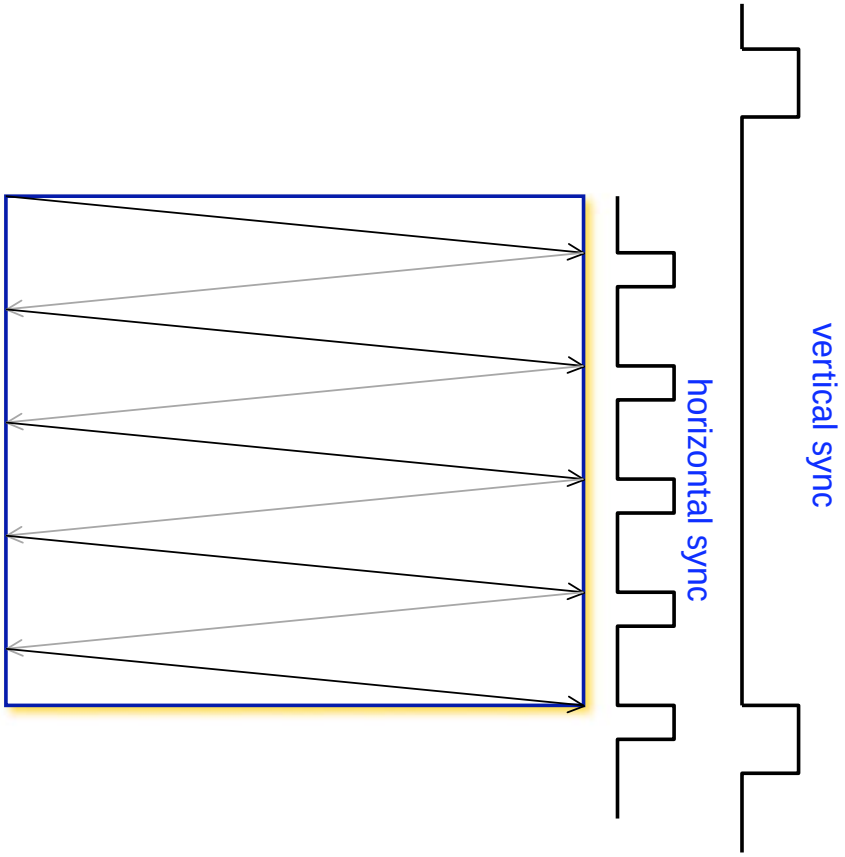


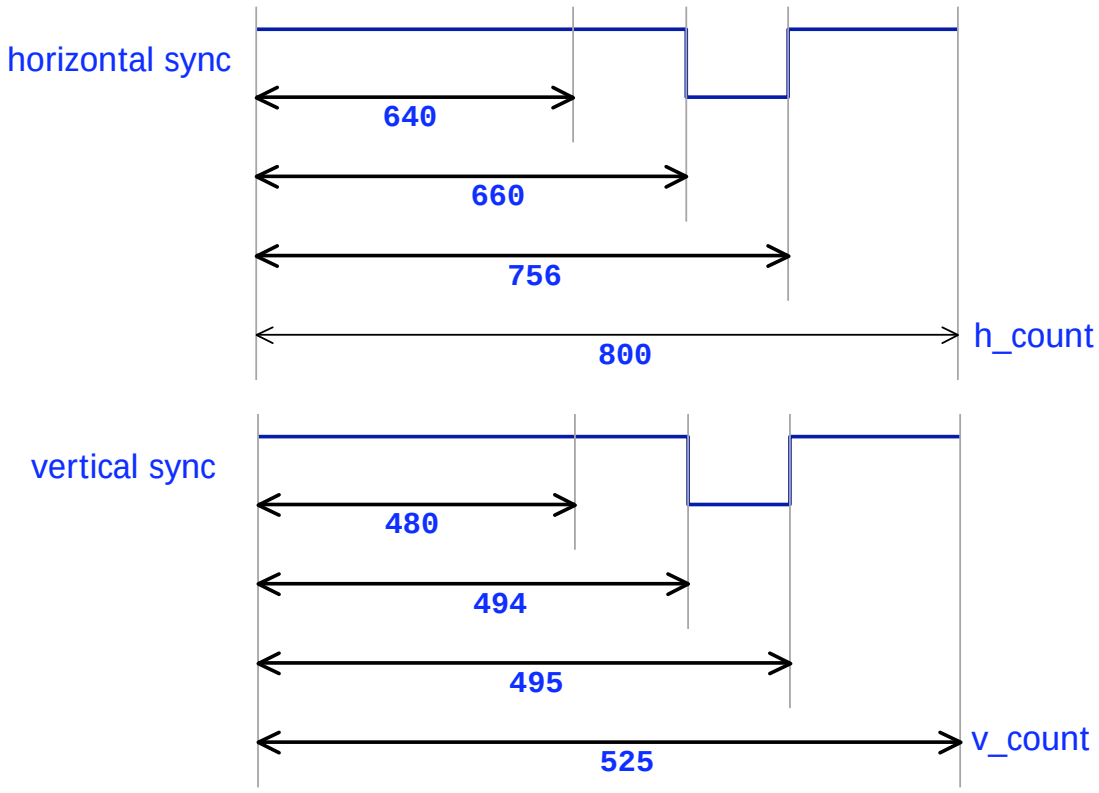
Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

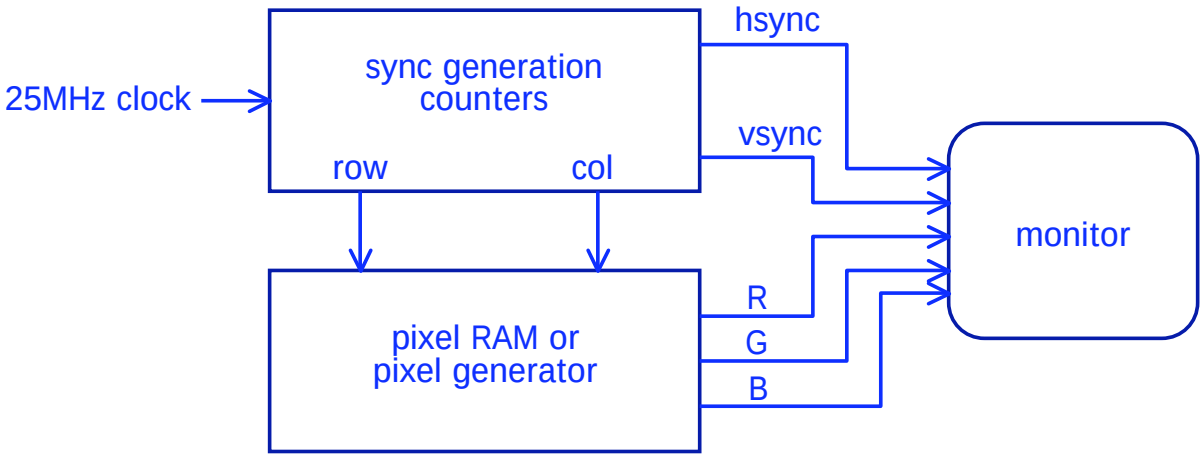
- The screen refresh process begins in the top left corner and paints 1 pixel at a time from left to right. At the end of the first row, the row increments and the column address is reset to the first column. Once the entire screen has been painted, the refresh process begins again
- The video signal must redraw the entire screen 60 times per second to provide for motion in the image and to reduce flicker: this period is called the refresh rate. Refresh rates higher than 60 Hz are used in PC monitors
- In 640 by 480-pixel mode, with a 60 Hz refresh rate, this is approximately 40 ns per pixel. A 25 MHz clock has a period of 40 ns

Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

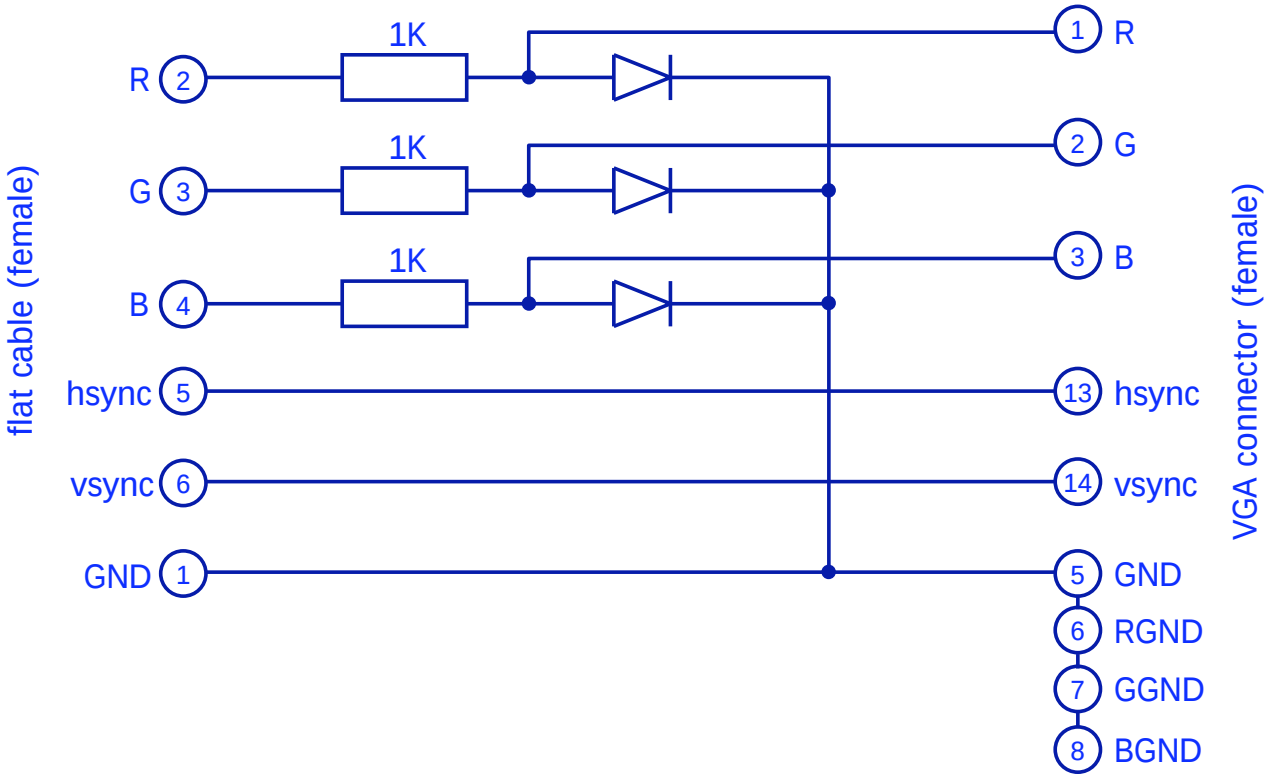
- The vertical sync signal tells the monitor to start displaying a new image or framme, and the monitor starts in the upper left corner with pixel (0,0)
- The horizontal sync signal tells the monitor to refresh another row of 640 pixels
- After 480 rows of pixels are refreshed with 480 horizontal sync signals, a vertical sync signal resets the monitor to the upper left corner and the process continues
- During the time when pixel data is not being displayed and the beam is returning to the left column to start another horizontal scan, the RGB signals should all be set to black color (all zero)
- In a PC graphics card, a dedicated memory location is used to store the color value of every pixel in the display. This memory is read out as the beam scanns across the screen to produce the RGB signals




horizontal sync
640
660
756
800
h_count
vertical sync
480
494
495
525
v_count


25MHz clock
sync generation
counters
row
col
hsync
vsync
monitor
pixel RAM or
pixel generator
R
G
B

VGA connector



Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity vga is
  port(clk, reset            : in std_logic;
       red, green, blue      : in std_logic;
       r, g, b, hsync, vsync : out std_logic;
       row                   : out std_logic_vector(8 downto 0);
       column                : out std_logic_vector(9 downto 0));
end vga;

architecture synt of vga is

  signal videoon, videov, videoh : std_logic;
  signal hcount, vcount          : std_logic_vector(9 downto 0);
```


Eduardo Sanchez
Ecole Polytechnique Fédérale de Lausanne

```
begin

hcounter: process (clk, reset)
          begin
          if reset='1'
            then hcount <= (others => '0');
            else if (clk'event and clk='1')
                   then if hcount=799
                          then hcount <= (others => '0');
                          else hcount <= hcount + 1;
                        end if;
                 end if;
          end if;
          end process;

          process (hcount)
          begin
          videoh <= '1';
          column <= hcount;
          if hcount>639
            then videoh <= '0';
                 column <= (others => '0');
          end if;
          end process;
```



```
vcounter: process (clk, reset)
          begin
          if reset='1'
            then vcount <= (others => '0');
            else if (clk'event and clk='1')
                   then if hcount=699
                          then if vcount=524
                                 then vcount <= (others => '0');
                                 else vcount <= vcount + 1;
                               end if;
                        end if;
                 end if;
          end if;
          end process;

          process (vcount)
          begin
          videov <= '1';
          row <= vcount(8 downto 0);
          if vcount>479
            then videov <= '0';
                 row <= (others => '0');
          end if;
          end process;
```

```
sync: process (clk, reset)
      begin
      if reset='1'
        then hsync <= '0';
             vsync <= '0';
        else if (clk'event and clk='1')
               then if (hcount<=755 and hcount>=659)
                      then hsync <= '0';
                      else hsync <= '1';
                    end if;
                    if (vcount<=494 and vcount>=493)
                      then vsync <= '0';
                      else vsync <= '1';
                    end if;
             end if;
      end if;
      end process;

videoon <= videoh and videov;
```



```
     colors: process (clk, reset)
             begin
             if reset='1'
               then r <= '0';
                    g <= '0';
                    b <= '0';
               elsif (clk'event and clk='1')
                 then r <= red and videoon;
                      g <= green and videoon;
                      b <= blue and videoon;
             end if;
             end process;

     end synt;
```